Ensemble of winning tickets: pruning bidirectional encoder from the transformers attention heads for enhanced model efficiency

Nyalalani Smarts, Rajalakshmi Selvaraj, Venu Madhav Kuthadi
Department of Computing and Informatics, School of Pure and Applied of Sciences, Botswana International University of Science and Technology, Palapye, Botswana

Article Info

Article history:

Received Jul 8, 2024
Revised Oct 15, 2024
Accepted Oct 23, 2024

Keywords:

Attention heads
Bidirectional encoder from the transformers
Lottery tickets hypothesis
Natural language processing
Pruning

ABSTRACT

The advanced models of deep neural networks like bidirectional encoder from the transformers (BERT) and others, poses challenges in terms of computational resources and model size. In order to tackle these issues, techniques of model pruning have surfaced as the most useful methods in addressing the issues of model complexity. This research paper explores the concept of pruning BERT attention heads across the ensemble of winning tickets in order to enhance the efficiency of the model without sacrificing performance. Experimental evaluations show how effective the approach is, in achieving significant model compression while still maintaining competitive performance across different natural language processing tasks. The key findings of this study include model size that has been reduced by 36%, with our ensemble model reaching greater performance as compared to the baseline BERT model on both Stanford Sentiment Treebank v2 (SST-2) and Corpus of Linguistic Acceptability (CoLA) datasets. The results further show a F1-score of 94% and 96%, respectively, and accuracy scores of 95% and 96% on the two datasets. The findings of this research paper contribute to the ongoing efforts in enhancing the efficiency of large-scale language models.

Corresponding Author:

Nyalalani Smarts
Department of Computing and Informatics, Faculty of Sciences, Botswana International University of Science and Technology
Plot 10071, Palapye, Botswana
Email: Nyalalani.smarts@studentmail.biust.ac.bw

1. INTRODUCTION

Deep neural networks like bidirectional encoder from the transformers (BERT) [1] have totally changed natural language processing (NLP) tasks. Previous research in the literature have demonstrated that BERT is a powerful model for NLP tasks. However, the alarming rate at which the models are growing in terms of model complexity with billions of parameters [2], [3] has lead to increased demand for computational resources and the large model sizes needed for training and deployment [4], [5]. The complications of the models make the practicality and scalability of these models in real-world applications to be difficult. Thus, there is a pressing need for more efficient and effective approaches to NLP [6].

Previous studies like those by [7]–[9] have shown BERT capabilities in regard to NLP tasks, with the another study by Maruf *et al.* [10] demonstrating its significant improvements in machine translation models for document-level translation, the study [11] illustrating its capabilities in contextual understanding and the study by Zakraoui *et al.* [12] in neural machine translation (NMT). Information from the literature shows that BERT has also excelled in sentiment analysis tasks and has proved its importance in various practical applications. Studies demonstrate that it has been applied to analyze customer reviews and feedback on e-commerce platforms [13]. It has also been used to analyze user generated content related to product and service evaluations [14] and other NLP tasks like question answering and text classification. Regardless of these achievements, these studies which highlighted above also note the challenges posed by BERT's large size and high computational requirements. Consequently, these indicate that methods that can reduce the complexity of models like BERT without sacrificing performance are needed. Currently, the pruning techniques that are available usually lead to a trade-off between model size and accuracy, which makes them less effective. This study aims to address these challenges by developing a more efficient pruning method.

We introduce a new pruning technique that we call ensemble of winning tickets which utilizes the lottery ticket hypothesis to enhance BERT’s performance metrics while drastically reducing its complexity. In this proposed approach, we implement an iterative attention head pruning for BERT retaining the minimal features of the model. In our approach the combined pruned subnetworks or winning tickets are combined into a robust ensemble model which gives up to 36% reduction in model size without deteriorating the performance.

Pruning techniques have become important in overcoming the challenges of increasing model complexity in deep neural networks. By selectively removing redundant or less impactful components, pruning streamlines network structures, improving inference speed, reducing memory usage, and enhancing computational efficiency [15]–[17]. One promising approach is the concept of “winning tickets,” which identifies efficient subnetworks within sparse deep neural networks that match or exceed the performance of the original model [18], [19].

In this study we make the following key contributions: i) Development of an ensemble method for pruning deep learning models; ii) Iterative pruning of BERT attention heads to retain only the most critical components; and iii) Integration of pruned subnetworks into a robust ensemble, resulting in a 36% reduction in model size without loss of performance. The remainder of this paper is structured as: section 2 describes the proposed ensemble of winning tickets method in detail. Section 3 presents experimental results demonstrating the efficacy of our approach and discusses the implications of our findings and potential future work. Finally, section 4 concludes the paper.

2. METHOD

Our approach includes the utilization of a well-established Stanford Sentiment Treebank (SST-2) [20] and Corpus of Linguistic Acceptability (CoLA) [21] datasets with training and development sets, the careful initialization of BERT-base lowercase model from the transformers library [22]. The approach also uses the innovative use of ensemble techniques, meticulously chosen training configurations and evaluated using the metrics outlined in Table 1. Each step is designed to optimize the learning process and enhance model efficiency, providing a comprehensive framework.

The proposed method general architecture is shown in Figure 1 which demonstrates how this study optimizes BERT model using pruned subnetworks, called winning tickets. Starting with a fully trained BERT model, iterative pruning removes unnecessary parameters, identifying efficient subnetworks that retain performance. These smaller BERT models called winning tickets are combined into an ensemble [23], enhancing efficiency and performance over the original model. The implementation of the ensemble pruning technique for BERT attention heads involves selectively identifying and retaining informative attention heads based on the principles of the lottery ticket hypothesis.

Table 1. CoLA, SST-2 tasks, dataset sizes, and metrics for this study

Task	Dataset	Train	Dev	Metrics	
Linguistic acceptability	Corpus of Linguistic Acceptability Judgements	10k	1k	Accuracy	F1-score
Sentiment analysis	The Stanford Sentiment Treebank	67k	872	Accuracy	F1-score

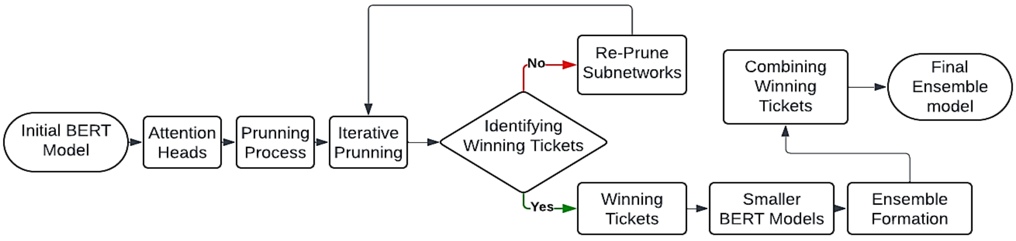


Figure 1. Steps of optimizing BERT models through an ensemble of pruned subnetworks

2.1. Ensemble pruning methods

Recent studies have explored various methods for leveraging multiple pruned subnetworks to enhance model performance and robustness, often through the lens of the lottery ticket hypothesis and ensemble techniques [24]. Another study by Chen *et al.* [25] introduced structured winning tickets at initial BERT training and Kobayashi *et al.* [26] discovered four winning tickets during the fine-tuning process. Furthermore, Gong *et al.* [27] regularized the process of fine-tuning using a technique called L1 distance and explored the structure of the discovered subnetworks.

Another example of the implementation of this technique can be seen in the study [28], in their research they applied the lottery ticket hypothesis to prune attention heads in machine translation. In this study we use the lottery ticket hypothesis to iteratively prune the attention heads, identifying winning tickets by retaining only the most critical attention heads from a BERT model. This method refines the subnetwork structure iteratively and integrates these pruned subnetworks into a robust ensemble.

2.2. Dataset and preprocessing

We utilized the SST-2 and CoLA datasets from the general language understanding evaluation (GLUE) benchmark for our experiments, dividing and shuffling the datasets using different random seeds to ensure that the model see the data in different order, with the BERT model checkpoint “bert-base-uncased” employed. The SST-2 dataset is a binary sentiment classification (positive/negative sentiment) dataset used for sentence-level sentiment analysis. The dataset does not need to be processed and is widely used in NLP tasks, ensuring that sentences are well-formatted and labels are consistent. For the SST-2 dataset, the training set was slightly reduced to 40,000 shuffled samples, the validation set comprised 800 samples, and the test set included 1,500 samples reason being to allow fast experimentation, to minimize the requirements of computational resources, including memory, processing power, and time. For the CoLA dataset which is used for linguistic acceptability classification to tests a model’s understanding of linguistic rules. The CoLA dataset is well-organized, with grammatically acceptable and unacceptable sentences already labeled and has been done by linguistic experts, to ensure quality and reliable data. For our experiments the training set contained 8,000 shuffled samples, the validation set consisted of 1,000 samples, and the test set had 1,000 samples. This specific splitting and shuffling ensured that our model training and evaluation were based on diverse and representative data samples.

2.3. Model and tokenizer initialization

We initialized the BERT model for sequence classification and the corresponding tokenizer using pre-trained checkpoints. This step ensures that our model benefits from transfer learning, leveraging the vast knowledge encoded in the BERT architecture. The BERT model and *AutoTokenizer* from the hugging face transformers library associated with *bert-base-uncased* called *BertTokenizer*, were carefully chosen to align with our experimental requirements, enabling efficient and accurate sequence classification tasks.

2.4. Ensemble of winning tickets

An empty ensemble set was initialized to store the winning tickets. To facilitate the pruning of attention heads, a mask matrix with all values set to 1 were created. This mask matrix is crucial for identifying and retaining the most critical components of BERT architecture. A sub-network (ticket) was randomly initialized based on the BERT architecture, using the specified checkpoint. The optimizer and loss function for the sub-network were set to Adam and cross-entropy loss, respectively. These configurations allowed for effective optimization and training of the sub-network, ensuring its performance aligns with the overall model goals.

2.5. Training configuration

The training parameters were meticulously chosen to optimize the learning process. We set the number of training iterations to five, with a learning rate of 2e-5 and a batch size of 16. A convergence threshold of 0.001 was established to ensure stable training. The training arguments included parameters for saving the model, loading the best model at the end of training, setting the number of training epochs to three, and defining the evaluation and saving strategies. The metrices for selecting the best model was set to the F1-score, accuracy and the total number of saved models was limited to 10. These configurations were designed to ensure that our model training was both efficient and robust, preventing overfitting while maximizing performance.

2.6. Selection of informative attention heads using the lottery ticket hypothesis

A saliency score is a metric used to measure the importance or relevance of individual elements within a model, such as neurons, features, or attention heads, based on their contribution to the model's performance. Saliency scores are typically derived from gradients or other sensitivity measures computed with respect to the input features or parameters of the model [29]. In the case of attention-based models like BERT, saliency scores were computed for the attention heads, which represent the relevance of different parts of the input to the model's output [30].

Let H_i denote the ith attention head. The saliency score for attention head H_i denoted as S_i were computed based on its impact on the model's loss or output. We measured the change in the model's loss L when H_i is perturbed or masked:

$$S_i = L(f(x, mask(H_i))) - L(f(x)) \tag{1}$$

Here, mask H_i represents a modification of the model where the ith attention head is masked or zeroed out.

2.7. Practical calculations and algorithms for identifying winning tickets

In this subsection, we specify the steps in detail and the algorithm that was used to carry out the ensemble of winning tickets, which particularly helps in reducing the number of attention heads of BERT more efficiently without compromising on the performance of the model. The main goal is to train the subnets of the BERT model and prune them, searching for possible subnets where their performance can be equal or higher than the performance of the original model. Here we describe methods which make use of independent pruned models to enhance the final model performance by using model ensemble technique. The Algorithm 1 given below presents a step-by-step strategy for establishing and creating these winning tickets leveraging both training and validation datasets to ensure optimal performance. The process leads up to aggregating predictions from multiple pruned subnetworks, using simple averaging to generate robust and reliable final predictions. This method aims for better performance in terms of computational time by providing normalization and enhancing the effectiveness of the large models, such as BERT.

Algorithm 1. Ensemble of winning tickets approach for BERT

```
Input:
    – A pre-trained BERT model: $M$
    – Number of ensembles to be created: $n$
    – Validation dataset: $D_v$
    – Test dataset: $D_t$
Output:
    – An ensemble of winning tickets: $T$
  Steps 1. Load the pre-trained BERT model: $M$
  Steps 2. Initialize an empty ensemble set to store the winning tickets: $T = \{\}$
  Steps 3. Set the number of ensembles to be created: $n$
  Steps 4. For each ensemble iteration $i$ from 1 to $n$:
          a. Initialize a mask matrix for the attention heads with all values set to 1: $M_i = 1_m \times d$
          b. Randomly initialize a sub-network (ticket) based on the BERT architecture:
             $T_i = random\_init(M_i)$
          c. Train the sub-network for a fixed number of iterations or until convergence on a training
             dataset: $T_i = train(T_i, D_t)$
          d. Evaluate the sub-network's performance on the validation dataset: $T_i = evaluate(T_i, D_t)$
          e. If the sub-network meets the desired performance criteria:
             – Apply the mask matrix to the sub-network, effectively pruning the selected attention
               heads: $T_i = prune(T_i, M_i)$
             – Add the pruned sub-network (winning ticket) to the ensemble set: $T = T \cup T_i$
  Steps 5. For each sample $x$ in the test dataset:
          a. Feed the sample through each winning ticket in the ensemble set: $P_i = T_i(x)$
          b. Aggregate the predictions from each winning ticket to obtain the final prediction:

             $P(x) = f(\theta_1(x), \theta_2(x), \dots, \theta_n(x)) = \frac{1}{n}\sum_{i=1}^{n} \theta_i(x)$

  Steps 6. Evaluate the performance of the ensemble on the test dataset using appropriate metrics:
           $M = evaluate(T, D_t)$
  Steps 7. Repeat steps 4 to 6 for the desired number of iterations or ensembles.
  Steps 8. Aggregate the performance metrics across all ensembles to assess the overall performance of the
           ensemble method: $M = aggregate\_metrics(T)$.
```

a. Notation

- Ensemble of winning tickets E: combining multiple pruned models M_i where $i = 1,2,\dots,n$.

$$E(x) = f(T_1(x), T_2(x), \dots, T_n(x)) \tag{2}$$

- Aggregation function f: aggregates the predictions of the pruned models using simple averaging.

$$P(x) = f(\theta_1(x), \theta_2(x), \dots, \theta_n(x)) = \frac{1}{n}\sum_{i=1}^{n} \theta_i(x) \tag{3}$$

b. Process of forming an ensemble

- Form the ensemble set T:

$$T = i = \bigcup_{i=1}^{n} (prune(\theta_i, 1) \mid evaluate(\theta_i, D_v)) \tag{4}$$

where:

$$\theta_i = train(init(1), D_t) \tag{5}$$

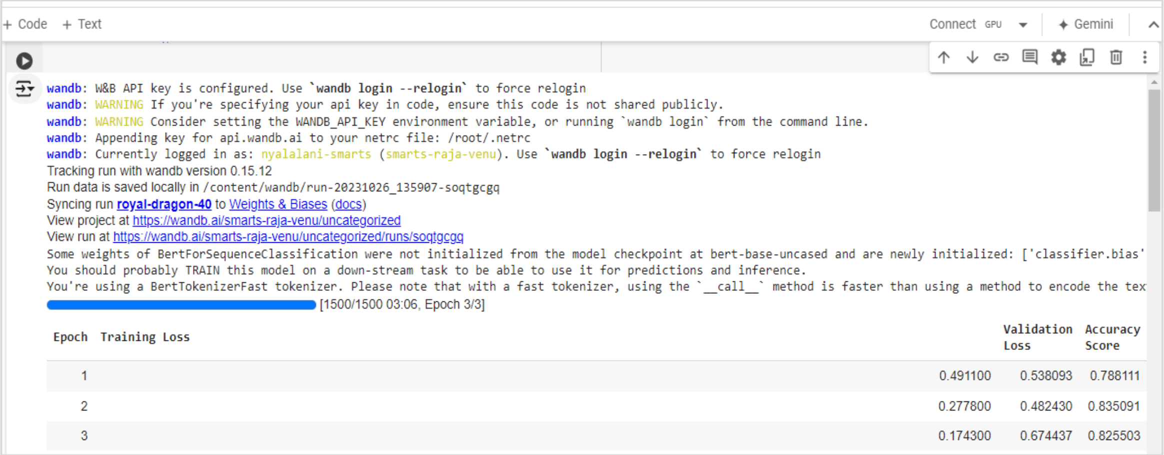
- Make predictions with aggregation:
  Obtain individual predictions from each pruned sub-network:

$$\theta_i(x) for\ i = 1,2,\dots,n \tag{6}$$

Aggregate these predictions using the aggregation function f:

$$P(x) = f(\theta_1(x), \theta_2(x), \dots, \theta_n(x)) = \frac{1}{n}\sum_{i=1}^{n} \theta_i(x) \tag{7}$$

The steps outlined in the algorithm above demonstrate the process of creating an ensemble from pruned attention heads of a BERT model and using this ensemble for aggregated predictions. The chosen methods ensure the reliability and reproducibility of results. The SST-2 and CoLA datasets from the GLUE benchmark provides a robust foundation for our study, while the BERT model provides a standard in sequence classification tasks, benefits from pre-trained checkpoints to enhance performance. The ensemble of winning tickets approach aims to improve model efficiency by focusing on the most crucial components of the BERT architecture. The training configurations are designed to optimize the learning process while preventing overfitting, ensuring the robustness of our findings. Figure 2 shows the simulation of the ensemble model during the training process. Our implementation codes can be found at GitHub: *https://github.com/nksmarts/Ensemble-of-winning-tickets/blob/main/New_pruning_BERT_attention_heads_F1_score.ipynb.*



```
wandb: W&B API key is configured. Use `wandb login --relogin` to force relogin
wandb: WARNING If you're specifying your api key in code, ensure this code is not shared publicly.
wandb: WARNING Consider setting the WANDB_API_KEY environment variable, or running `wandb login` from the command line.
wandb: Appending key for api.wandb.ai to your netrc file: /root/.netrc
wandb: Currently logged in as: nyalalani-smarts (smarts-raja-venu). Use `wandb login --relogin` to force relogin
Tracking run with wandb version 0.15.12
Run data is saved locally in /content/wandb/run-20231026_135907-soqtgcgq
Syncing run royal-dragon-40 to Weights & Biases (docs)
View project at https://wandb.ai/smarts-raja-venu/uncategorized
View run at https://wandb.ai/smarts-raja-venu/uncategorized/runs/soqtgcgq
Some weights of BertForSequenceClassification were not initialized from the model checkpoint at bert-base-uncased and are newly initialized: ['classifier.bias'
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
You're using a BertTokenizerFast tokenizer. Please note that with a fast tokenizer, using the `__call__` method is faster than using a method to encode the tex
[1500/1500 03:06, Epoch 3/3]
```

Epoch	Training Loss	Validation Loss	Accuracy Score
1	0.491100	0.538093	0.788111
2	0.277800	0.482430	0.835091
3	0.174300	0.674437	0.825503

Figure 2. Simulation of ensemble model during the training process

3. RESULTS AND DISCUSSION

This section presents a detailed overview of our experiments, results, and discussions, focusing on the performance and efficiency of the BERT model as compared to our ensemble model on the CoLA and SST-2 datasets. The analysis includes a performance comparison of the pruned ensemble models against the baseline models, offering insights into the efficiency-performance trade-off. Furthermore, we investigate the effects of pruning on accuracy, inference speed, and computational expenses, pointing out the benefits and shortcomings of using pruned models. The results emphasize on the advantage of the ensemble method to bring out an improved performance in efficiency without sacrificing the task performance.

3.1. Experiments setup

We start with the BERT base-uncased model, featuring 12 layers, 12 attention heads, 110 million parameters, and 768 hidden units per layer. Using different random seeds, we evaluate with F1-score and accuracy. Training parameters include a batch size of 16, learning rate of 2e-5, 5 iterations, and a convergence threshold of 0.001, we use Google Collab a cloud service on a single NVIDIA Tesla T4 GPU. We use the Adam optimizer and cross-entropy loss function to update parameters and minimize loss.

3.2. Performance comparison

The effectiveness of our ensemble model is evident in its performance on the CoLA and SST-2 datasets, as shown in Table 2. We compare our ensemble model to BERT baseline model, EarlyBERT, and dominant winning ticket using the default training settings for pre-training and fine-tuning. Our ensemble achieves superior accuracy and F1-scores of 95% and 94% on CoLA, and 96% and 96% on SST-2. These results indicate the efficiency gains of our approach, balancing the higher performance of the model while managing its complexity.

Our findings align with previous studies which demonstrate that ensemble methods have shown promise in enhancing model performance across various tasks by leveraging the strengths of multiple models [25], particularly with [26], which also identified four subnetworks during the fine-tuning process. The consistent outperformance of our ensemble model over other baselines indicates how important an ensemble technique is in natural language processing. The substantial reduction in parameters from 110 million in the BERT baseline to 70 million after pruning shows that pruning not only enhances efficiency but also improves generalization across datasets. Our ensemble model offers a compelling key for tasks requiring high accuracy and efficiency. The results presented support the hypothesis that combining pruning technique with ensemble methods can greatly boost deep learning model performance.

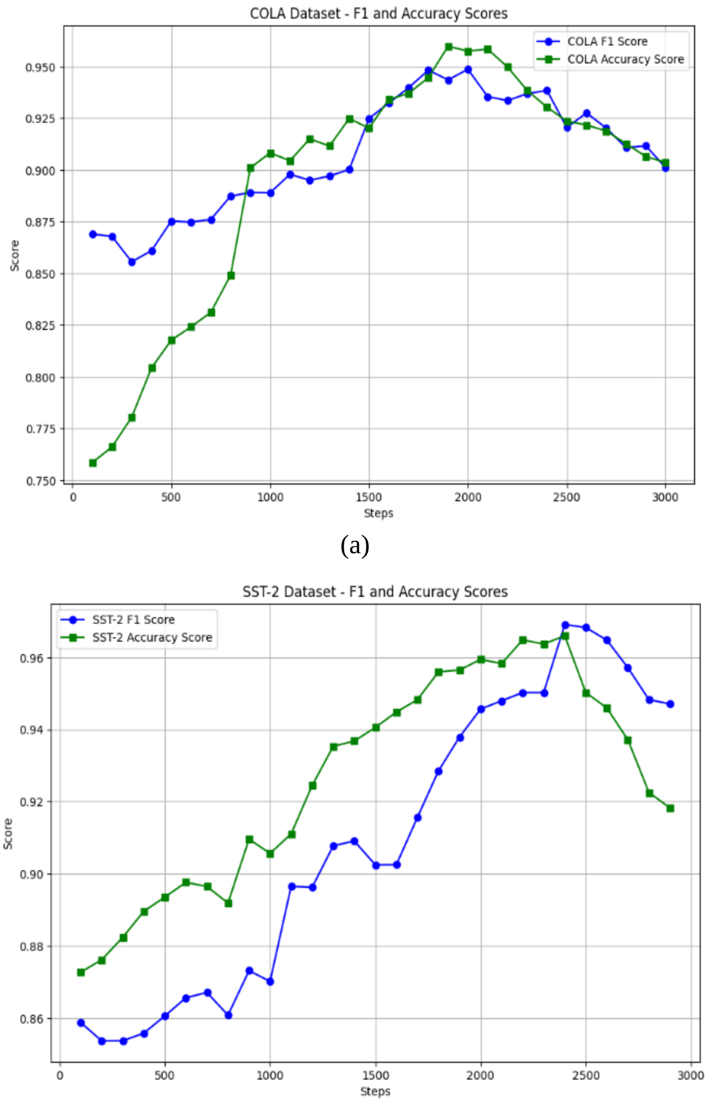
Table 2. Model performance comparison with other ensemble techniques

Model	Dataset	Metrics	Reference
BERT	CoLA	60.5	[1]
EarlyBERT	CoLA	52	[25]
Dominant winning ticket	CoLA	68	[27]
Our ensemble model	CoLA	95.0	-
BERT	SST-2	92.7	[1]
EarlyBERT	SST-2	90.71	[25]
Dominant winning ticket	SST-2	96.4	[27]
Our ensemble model	SST-2	96.0	-

3.3. Experimental findings on model performance trends

The performance of our ensemble model, as shown in Figures 3, highlights key findings in its application to the CoLA and SST-2 datasets. Specifically, on the CoLA dataset Figure 3(a), the F1-score exhibits an initial steady increase, peaking at 1700 and 2000 training steps before gradually declining, whereas accuracy improves significantly after 800 steps, reaching its peak at 1900 steps with minor fluctuations thereafter. In contrast, the SST-2 dataset in Figure 3(b) reveals a more consistent performance, with the F1-score and accuracy both showing continuous improvements, peaking at around 2400 and 2300 steps, respectively. The model's superior performance on the SST-2 dataset, with an F1-score of 96%, underscores its effectiveness in sentiment analysis tasks, especially when compared to its results on the more challenging, imbalanced CoLA dataset, nevertheless our ensemble model has shown a remarkable performance with a score of 95% compared to other models which have far less performance.

When compared to previous studies, such as those highlighting the capabilities of BERT in natural language understanding [13], our ensemble model's performance aligns well, particularly in sentiment analysis. However, the model's performance on the CoLA dataset suggests that while it is highly effective, it may require additional optimizations to fully address the challenges posed by imbalanced datasets. The study's strength lies in demonstrating the model's adaptability across different tasks, although the need for task-specific adjustments, such as early stopping during training on CoLA, highlights a limitation in its generalizability without fine-tuning. Unexpectedly, the consistent improvement in SST-2 performance suggests that the model is particularly well-suited for sentiment analysis, possibly due to the dataset's balanced nature compared to CoLA. The study demonstrates the effectiveness of our ensemble model in enhancing performance across sentiment analysis and linguistic acceptability in natural language understanding tasks, with particularly impressive results in sentiment analysis as evidenced by the performance metrics. The findings highlight the importance of task-specific optimizations and ensembling techniques in boosting the performance of the model.



(a)

(b)

Figure 3. Metrics over steps: (a) CoLA dataset-F1-score and accuracy scores and (b) SST-2 dataset-F1 and accuracy scores

3.4. Model efficiency

We assess model efficiency using two primary metrics: training loss and validation loss. The training loss decreases steadily across epochs, which indicate effective learning and adaptation to the training data's complexities. In this study, both runs (Run 1 and Run 2) represented in Figure 4 shows a significant decrease in training loss over epochs, demonstrating effective learning. Run 1 starts with a slightly lower initial training loss (0.4794) compared to Run 2 (0.4871), with both converging to similar values by the third epoch (0.1679 for Run 1 and 0.1783 for Run 2). The consistent reduction of training loss shows that the model is effectively fitting the training data.

When comparing validation losses, Run 1 shows a steady decrease from 0.439961 to 0.229722, suggesting good generalization. In contrast, Run 2 shows an initial decrease from 0.462198 to 0.301459, followed by a slight increase to 0.299997. This differences hints that the model could be potentially overfitting or there's variability in model performance. The lower validation loss in Run 1 suggests it generalizes slightly better, showing the importance of stable validation performance for model robustness. The study confirms that the model learns and generalizes well, with decreasing training and validation losses. Monitoring these metrics offers insights into the model's generalization ability to unseen data and guides adjustments such as regularization techniques or modifications to the model architecture to improve performance and prevent overfitting.

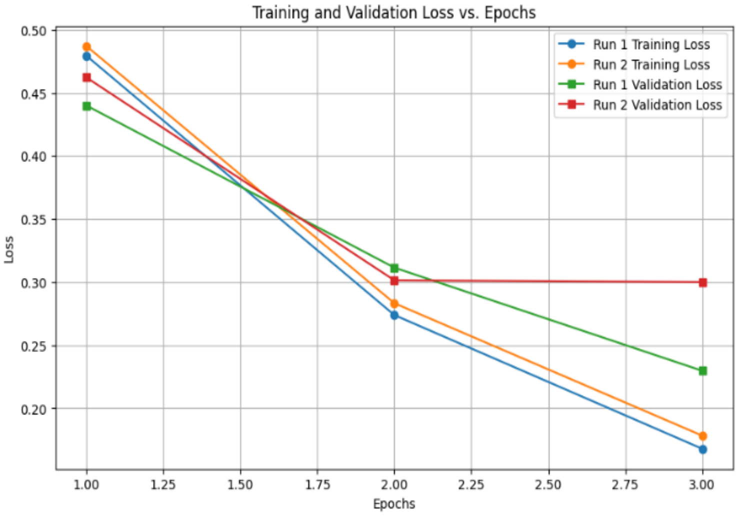


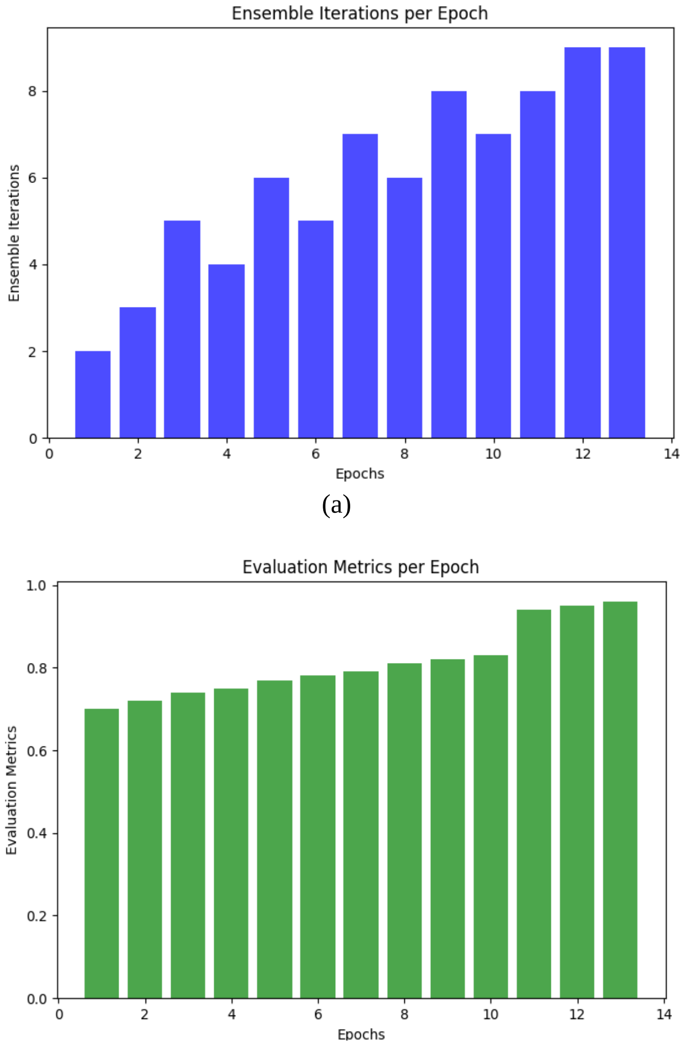
Figure 4. Training and validation loss vs epochs

Figure 5 shows the development of ensemble iterations and evaluation metrics across epochs, providing a visual representation of: (a) the number of ensemble iterations executed per epoch and (b) the corresponding evaluation metrics achieved at each epoch. Figure 5(a) shows the fluctuating number of ensemble iterations per epoch, depicted by blue bars. Figure 5(b) displays the upward trend in evaluation metrics across epochs, represented by green bars. Figures 5(a) and 5(b) shows insightful observations regarding the efficiency and performance of our model utilizing ensemble pruning techniques across 13 epochs. Remarkably, training and fine-tuning the model per 3 epochs took about 45 minutes to 1 hour, although more than 5 hours was needed to fine-tune using 13 epochs.

A comparison can be drawn between the study of [24] utilizing up to 20 epochs to fine-tune their model, whereas in our study, we used only 3 and 13 epochs and still our ensemble model achieved outstanding performance. Therefore, the increase in ensemble iterations over epochs (ranging from 2 to 9) aligns with findings by Zheng *et al.* [24], who noted that ensemble methods can improve the performance of the model by leveraging multiple learning algorithms to yield improved prediction accuracy. Furthermore, in this study, we focused on optimizing the pruning process within an ensemble model by targeting attention heads, employing a fixed learning rate of 2e-5, and setting the pruning threshold to 0.001. Our approach allowed for the systematic reduction of the model's complexity, particularly by pruning entire blocks or structures of attention heads, resulting in a streamlined model architecture.

Comparatively, previous studies like that of [24]–[27] used iterative magnitude pruning, which focuses on individual weights. This technique allows for more fine grain pruning but can end up with a less reduction in model size as compared to our approach. While our method is focused in pruning larger structures within the model, likely leading to more reductions in model complexity, it may also introduce more variability in model performance. Previous studies focused on using weight pruning with a smaller threshold which provides finer control over model pruning and might better preserve model accuracy. The different focus areas of pruning which are weights vs. attention heads show the trade-offs between precision and simplifying the structural of the model.

The primary purpose of this study was to explore the effectiveness of attention head pruning within ensemble models and its impact on model efficiency. Our findings draw attention to the potential for a significant model reduction through attention head pruning while maintaining performance, particularly within the context of an ensemble. This study highlights the importance of exploring alternative pruning strategies, such as block-level pruning, and invites future research to delve deeper into the effects of different threshold levels and learning rates on model performance.



(a)

(b)

Figure 5. Ensemble iterations and evaluation metrics: (a) ensemble iterations per epoch and (b) evaluation metrics per epoch

4. CONCLUSION

This paper introduces a novel approach to enhance BERT model efficiency by pruning attention heads using an ensemble of winning tickets. Experimental results demonstrate significant reductions in model complexity while maintaining or improving performance across various NLP tasks. Our experiments on CoLA and SST-2 datasets show the pruned ensemble model reduces parameters from 110 million to 70 million, achieving superior accuracy and F1-scores compared to the original BERT. Specifically, our ensemble model achieved 95% accuracy and 94% F1-score on SST-2, and 96% for both metrics on CoLA. This underscores the efficacy of pruning informative attention heads to optimize model performance efficiently. Our analysis also emphasizes selecting appropriate evaluation metrics tailored to dataset characteristics; F1-score for CoLA's imbalanced data and accuracy for sentiment analysis in SST-2. In conclusion, the ensemble of winning tickets offers a promising avenue to optimize large-scale language models, balancing efficiency and performance, with potential applications across diverse NLP tasks and beyond. Future research could explore extending these techniques to other NLP tasks or investigating the effects of different pruning strategies on ensemble models. Further investigation into the variability in performance across different pruning techniques could offer valuable insights for optimizing model efficiency and robustness in various deep learning applications.

ACKNOWLEDGEMENTS

The authors are grateful to Botswana International University of Science and Technology for its support.

REFERENCES

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North*, Stroudsburg: Association for Computational Linguistics, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

[2] A. Tang, P. Quan, L. Niu, and Y. Shi, "A survey for sparse regularization based compression methods," *Annals of Data Science*, vol. 9, no. 4, pp. 695–722, Aug. 2022, doi: 10.1007/s40745-022-00389-6.

[3] C. Yu, T. Chen, and Z. Gan, "Boost transformer-based language models with GPU-friendly sparsity and quantization," in *Findings of the Association for Computational Linguistics: ACL 2023*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2023, pp. 218–235, doi: 10.18653/v1/2023.findings-acl.15.

[4] C.-C. J. Kuo and A. M. Madni, "Green learning: introduction, examples and outlook," *Journal of Visual Communication and Image Representation*, vol. 90, pp. 103685, Feb. 2023, doi: 10.1016/j.jvcir.2022.103685.

[5] R. Verdecchia, J. Sallou, and L. Cruz, "A systematic review of green AI," *WIREs Data Mining and Knowledge Discovery*, vol. 13, no. 4, Jul. 2023, doi: 10.1002/widm.1507.

[6] J. D. Nunes, M. Carvalho, D. Carneiro, and J. S. Cardoso, "Spiking neural networks: a survey," *IEEE Access*, vol. 10, pp. 60738–60764, 2022, doi: 10.1109/ACCESS.2022.3179968.

[7] S. Shreyashree, P. Sunagar, S. Rajarajeswari, and A. Kanavalli, "A literature review on bidirectional encoder representations from transformers," in *Inventive Computation and Information Technologies: Proceedings of ICICIT 2021*, Springer, 2022, pp. 305–320.

[8] W. Khan, A. Daud, K. Khan, S. Muhammad, and R. Haq, "Exploring the frontiers of deep learning and natural language processing: a comprehensive overview of key challenges and emerging trends," *Natural Language Processing Journal*, vol. 4, pp. 100026, Sep. 2023, doi: 10.1016/j.nlp.2023.100026.

[9] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: state of the art, current trends and challenges," *Multimedia Tools and Applications*, vol. 82, no. 3, pp. 3713–3744, Jan. 2023, doi: 10.1007/s11042-022-13428-4.

[10] S. Maruf, F. Saleh, and G. Haffari, "A survey on document-level neural machine translation," *ACM Computing Surveys*, vol. 54, no. 2, pp. 1–36, Mar. 2022, doi: 10.1145/3441691.

[11] A. Bandi, P. V. S. R. Adapa, and Y. E. V. P. K. Kuchi, "The power of generative AI: a review of requirements, models, input–output formats, evaluation metrics, and challenges," *Future Internet*, vol. 15, no. 8, pp. 260, Jul. 2023, doi: 10.3390/fi15080260.

[12] J. Zakraoui, M. Saleh, S. Al-Maadeed, and J. M. Alja'am, "Arabic machine translation: a survey with challenges and future directions," *IEEE Access*, vol. 9, pp. 161445–161468, 2021, doi: 10.1109/ACCESS.2021.3132488.

[13] M. S. Sayeed, V. Mohan, and K. S. Muthu, "BERT: a review of applications in sentiment analysis," *HighTech and Innovation Journal*, vol. 4, no. 2, pp. 453–462, Jun. 2023, doi: 10.28991/HIJ-2023-04-02-015.

[14] K. A. Alshaikh, O. A. Almatrafi, and Y. B. Abushark, "BERT-based model for aspect-based sentiment analysis for analyzing Arabic open-ended survey responses: a case study," IEEE Access, vol. 12, pp. 2288–2302, 2024, doi: 10.1109/ACCESS.2023.3348342.

[15] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang, "Pruning and quantization for deep neural network acceleration: a survey," *Neurocomputing*, vol. 461, pp. 370–403, Oct. 2021, doi: 10.1016/j.neucom.2021.07.045.

[16] B. Li, Y. Miao, Y. Wang, Y. Sun, and W. Wang, "Improving the efficiency and effectiveness for BERT-based entity resolution," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, pp. 13226–13233, doi: 10.1609/aaai.v35i15.17562.

[17] V. K. Verma, N. Mehta, S. Si, R. Henao, and L. Carin, "Pushing the efficiency limit using structured sparse convolutions," in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, IEEE, Jan. 2023, pp. 6492–6502, doi: 10.1109/WACV56688.2023.00644.

[18] Y. Yang, C. Zhang, B. Wang, and D. Song, "Doge tickets: uncovering domain-general language models by playing lottery tickets," in *CCF International Conference on Natural Language Processing and Chinese Computing*, 2022, pp. 144–156, doi: 10.1007/978-3-031-17120-8_12.

[19] Z. Gan *et al.*, "Playing lottery tickets with vision and language," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, pp. 652–660, doi: 10.1609/aaai.v36i1.19945.

[20] M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5731–5780, Oct. 2022, doi: 10.1007/s10462-022-10144-1.

[21] A. Warstadt, A. Singh, and S. R. Bowman, "Neural network acceptability judgments," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 625–641, Nov. 2019, doi: 10.1162/tacl_a_00290.

[22] T. Wolf *et al.*, "Transformers: state-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 38–45, doi: 10.18653/v1/2020.emnlp-demos.6.

[23] A. Mohammed and R. Kora, "A comprehensive review on ensemble deep learning: Opportunities and challenges," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 2, pp. 757–774, Feb. 2023, doi: 10.1016/j.jksuci.2023.01.014.

[24] R. Zheng *et al.*, "Robust lottery tickets for pre-trained language models," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2022, pp. 2211–2224, doi: 10.18653/v1/2022.acl-long.157.

[25] X. Chen, Y. Cheng, S. Wang, Z. Gan, Z. Wang, and J. Liu, "EarlyBERT: efficient BERT training via early-bird lottery tickets," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2021, pp. 2195–2207, doi: 10.18653/v1/2021.acl-long.171.

[26] S. Kobayashi, S. Kiyono, J. Suzuki, and K. Inui, “Diverse lottery tickets boost ensemble from a single pretrained model,” in *Proceedings of BigScience Episode #5 -- Workshop on Challenges & Perspectives in Creating Large Language Models*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2022, pp. 42–50, doi: 10.18653/v1/2022.bigscience-1.4.

[27] Z. Gong *et al.*, “Finding the dominant winning ticket in pre-trained language models,” in *Findings of the Association for Computational Linguistics: ACL 2022*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2022, pp. 1459–1472, doi: 10.18653/v1/2022.findings-acl.115.

[28] M. Behnke and K. Heafield, “Losing heads in the lottery: pruning transformer attention in neural machine translation,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 2664–2674, doi: 10.18653/v1/2020.emnlp-main.211.

[29] A. Bitar, R. Rosales, and M. Paulitsch, “Gradient-based feature-attribution explainability methods for spiking neural networks,” *Frontiers in Neuroscience*, vol. 17, Sep. 2023, doi: 10.3389/fnins.2023.1153999.

[30] C. Yeh, Y. Chen, A. Wu, C. Chen, F. Viégas, and M. Wattenberg, “AttentionViz: a global view of transformer attention,” *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–11, 2023, doi: 10.1109/TVCG.2023.3327163.

BIOGRAPHIES OF AUTHORS

Nyalalani Smarts He is a Ph.D. candidate and holds a master's degree in information systems and data management from Botswana International University of Science and Technology (BIUST) and a bachelor's degree in multimedia from RMIT University, Melbourne. He has held teaching positions at various institutions, including Limkokwing University, BIUST, and the Oodi College of Applied Arts and Technology. His research interest is in deep learning, NLP, machine learning, big data analytics, and multimedia. Throughout his career, He has been dedicated to student success and comprehensive learning, utilizing instructional technologies to enhance both in-class and online education. He has also reviewed academic programs for accreditation and contributed to multiple research projects. He has a strong commitment to research and innovation. He can be contacted at email: nyalalani.smarts@studentmail.biust.ac.bw.

Rajalakshmi Selvaraj is an associate professor in the Department of Computer Science and Information Systems at the Botswana International University of Science and Technology (BIUST). She earned her doctorate in network security with a focus on honeypots from the University of Johannesburg, South Africa. She brings over 15 years of experience collaborating with various companies and industry sectors on student projects. Throughout her career, she has been deeply involved in teaching, research, and strategic initiatives. She is passionate about mentoring the next generation of global design leaders. Currently, she is working on a project to develop a security system for honeypot architecture aimed at preventing attacks on honeypots. Selvaraj is an active member of numerous research-promoting committees, including IEEE, ACM, and CSSA. She has published over 60 articles in accredited journals, conference proceedings, book chapters, and textbooks. Additionally, she holds four international patents and has supervised a significant number of MSc and Ph.D. students. She can be contacted at email: selvarajr@biust.ac.bw.

Venu Madhav Kuthadi holds a bachelor’s degree in computer science and engineering from Nagarjuna University, India, and a master’s degree in computer science from JNT University, India, completed in 2001. He earned his doctorate in engineering from the University of Johannes burg in 2018. Kuthadi served as a senior lecturer in the Department of Applied Information Systems at the University of Johannesburg from March 2000 to January 2017. He is currently an associate professor in the Department of Computer Science and Information Systems at the Botswana International University of Science and Technology (BIUST). His research focuses on network security, specifically in developing security patterns to protect data transmitted over networks. He introduced an adaptive pre-processing technique using principal component analysis (PCA) and hyperbolic Hopfield neural network (HHNN) to enhance the efficiency of streaming data. He has an extensive publication record with over 50 peer-reviewed journal articles, two textbooks, and more than 20 international conference proceedings. He has successfully supervised 10 master's students and 3 Ph.D. candidates. Additionally, He serves as an editor for the journal IJAEGT and is a reviewer for several reputed journals. He can be contacted at email: kuthadiv@biust.ac.bw.